LABORATORIO 8

Programas Listas estáticas y Tipos Mixtos - Básico

1. Implementa la función **comunes** que, dados dos vectores de enteros cuyos valores están ordenados crecientemente, devuelva cuántos elementos son comunes a los dos vectores. Como ejemplo, con (3, 4, 5, 6, 7, 8, **9**, 10, **11**, 12) y (**9**, **11**, 13, 15, 17, 19, 21, 23, 25, 27), devuelve **2** (el 9 y el 11 son comunes a los dos). Nota: El hecho de que los vectores estén ordenados debe influir en el algoritmo a utilizar y una solución con dos algoritmos anidados no es aceptable.
2. Implementa el procedimiento **Encriptar** que, dados dos vectores de enteros *N*, *Clave,* devuelva otro vector *Num* con los dígitos de *N* recolocados usando la clave. N representa los dígitos de un número entero y clave tiene los números de 1 a N'length. Para encriptar N, se coloca cada uno de los dígitos de N en la posición indicada en V_Clave, suponiendo siempre que el número N se rellena con tantos ceros a la izquierda como sea necesario. Por ejemplo, si N es (8,7,5,3,9) y Clave es (2,4,1,5,3), el resultado es (5,8,9,7,3), ya que el 8 va a la segunda posición, el 7 va a la cuarta posición, el 5 va a la primera posición, el 3 va a la quinta posición y el 9 va a la tercera posición. Si N es(5,7,6) y Clave es (2,4,1,5,3), el resultado es (5,0,6,0,7) ya que el (5,7,6) se considera como si tuviera cinco cifras (0,0,5,7,6) y luego se aplica el algoritmo de encriptación: el primer 0 va a la segunda posición, el segundo 0 va a la cuarta posición, el 5 va a la primera posición, el 7 va a la quinta posición y el 6 va a la tercera posición.
3. Implementa el procedimiento **Insertar_En_Medio** que, a partir de una lista estática y un número natural N, lo inserte en la mitad de la lista, si el número de elementos de la lista es par. Si el número de elementos de la lista es impar o la lista está llena se deja la lista como está. Por ejemplo, si L= [4, 5, 3, 6] y N= 9 debería quedar [4, 5, 9, 3, 6]. Si L= [4, 6, 3] y N= 9 debería quedar [4, 6, 3]. Si L= [] y N= 3 debería quedar [3].
4. Implementa el procedimiento **Borrar_Intermedio** que, dada una lista dinámica de enteros L, la modifique eliminando el elemento intermedio (aquel que tiene el mismo número de nodos antes y después de dicho elemento). Esto significa que el subprograma debe eliminar un elemento SOLO cuando L tiene un número impar de elementos. Si L tiene un número par de elementos, entonces no se hace nada a la lista de partida.

(continúa)

5. Implementa en Ada el procedimiento **Obtener_Num_Vecinos_Por_Vivienda** que, dados los datos de toda la comunidad, obtenga la distribución de los vecinos por vivienda en el rascacielos. Tenemos datos de los 3.546 vecinos de una comunidad de vecinos. Por cada vecino tenemos información de su nombre, número de piso y mano donde vive. La comunidad vive en un rascacielos de 100 pisos (numerados del 1 al 100) y en cada piso hay 10 manos (nombradas desde la 'A' a la 'J'). Para acceder más fácilmente al número de habitantes por cada vivienda (piso y mano concretos), queremos transformar los datos en otra estructura de datos (matriz T_Rascacielos). Por ejemplo, si en el 1º A viven 5 personas en el vector de la comunidad habrá 5 posiciones en las que aparecerá información de vecinos del piso 1 y mano A. Tras el proceso de ir revisando el vector de la comunidad de vecinos, en R (de tipo T_Rascacielos) aparecerá el valor 5 en la primera celda, es decir, R(1, A)=5. NOTA sobre eficiencia: el procedimiento debe resolver el problema con un único recorrido del vector de la Comunidad.

Jon **1** **A**	Aiora 3 B	Koldo 2 A	Miren 2 A		**Mireia** **1** **A**	Leire 99 A		**Iker** **1** **A**			**Aritz** **1** **A**			**Ane** **1** **A**		
1	2			...	32		....	145		...	1345		...	2500		....

	'A'	**'B'**		**...**	**'J'**
1	5				
2	2				
3		1			
4					
...					
99	1				
100					

6. Tenemos información de una comunidad de vecinos que vive distribuida en un rascacielos de 100 pisos (numerados del 1 al 100) y por cada piso hay 10 manos (nombradas de la 'A' a la 'J'). Por cada vivienda (piso y mano concretos) tenemos información del consumo de electricidad, consumo de gas y número de habitantes. Toda esa información la representamos con T_Edificio. Implementa el procedimiento **Obtener_Consumos** que, dada la información de todo el edificio, y **utilizando un único recorrido de la matriz**, obtenga, teniendo en cuenta el número de habitantes en cada piso: (1) el consumo eléctrico medio por habitante de todo el rascacielos (un único valor para todo el edificio, el cálculo usa todos los habitantes del edificio), y (2) los consumos medios de gas por habitante de cada mano del rascacielos (10 valores distintos, uno para cada mano; usa T_Consumo_Medio_Manos para el resultado; en cada cálculo solo se tienen en cuenta los habitantes de esa mano en el edificio).